A new digital image correlation algorithm for whole-field displacement measurement

C. SU and L. ANAND
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Abstract— **We have developed a new digital image correlation (DIC) algorithm for non-contact, two-dimensional, whole-field displacement and strain measurement. Relative to existing algorithms, our algorithm substantially reduces the calculation expense by using neighborhood information while processing the data to determine the displacement field in a sub-region of interest. The new algorithm also uses higher-order interpolations of the displacement field, allowing for better accuracy in estimating strain distributions when the deformation field is non-homogeneous. Numerically-generated digital images are used to show that the new algorithm accurately reproduces the imposed displacement fields. The algorithm is also tested on actual images from deformed specimens from a variety of experiments, and shown to perform satisfactorily.**

Keywords— **A. Experimental Mechanics B. Non-contact strain measurement. C. Mechanical testing**

I. Introduction

THE digital image correlation (DIC) method obtains the incremental displacement and strain field on the surface of a planar specimen by comparing a pair of digital images taken before and after the deformation. The idea of using digital images as a strain measurement technique, was first introduced by Peters *et al.* [1], and has since been widely used in experimental strain analysis [cf., e.g., 2-9].

As a strain measurement technique, the DIC method has many desirable properties: (a) It is non-contact, and is therefore suitable for measuring the deformation response of soft materials, such as polymeric and biomaterials. (b) It gives *whole-field* displacement and strain measurements, which is important when the deformation is non-homogeneous. (c) It is useful for strain measurement at the micro- and nanoscale from digital photomicrographs taken using techniques such as TEM, SEM, and AFM.

Commonly, images of a surface during deformation are recorded by digital photography. A digital image may be considered to be a matrix whose elements are positive integers which represent the surface brightness; each of these elements is called a pixel with a given value in a range which depends on the digital camera and the image acquisition electronics.[1] By comparing successive images the 2D in-plane incremental displacement field is measured.[2]

Consider an object with a flat surface, and let $\mathbf{X}$ be the position of a material point on the surface in the reference configuration at time t_0, and $\mathbf{x}$ its position in the deformed configuration of the surface at a later time $t > t_0$, then

$$\mathbf{x} = \mathbf{y}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t), \quad t > t_0, \tag{1}$$

where $\mathbf{y}$ is the motion, and $\mathbf{u}$ is the displacement vector of the material point $\mathbf{X}$. During an experiment, images of the surface of the object are recorded at times t_0 and t. Let $f(\mathbf{X}, t_0)$ and $g(\mathbf{x}, t)$ represent the intensity distribution functions of the two images in the initial and deformed configurations, respectively. The principle of image correlation for deformation measurement is to find a displacement field $\mathbf{u}(\mathbf{X}, t)$ that can match the two intensity distribution functions over a small but finite area S in the initial configuration, or the corresponding area s in the deformed configuration. The displacement field $\mathbf{u}$ over S is often parameterized by a vector $\mathbf{P}$

$$\begin{aligned} \mathbf{u}(\mathbf{X}, t) &= \mathbf{u}_{\mathbf{P}}(\mathbf{X}; \mathbf{P}), \\ \mathbf{P} &= \{P_1(\mathbf{X}_0, t), P_2(\mathbf{X}_0, t), \cdots, P_N(\mathbf{X}_0, t)\}, \end{aligned} \tag{2}$$

where $\mathbf{X}_0$ is a reference point within the subset S. The match between the two images over the subset S for an assumed parameter set $\mathbf{P}$ is measured by a *correlation coefficient*

$$C = C(\mathbf{P}) = \tilde{C}(f(\mathbf{X}, t_0), g(\mathbf{X} + \mathbf{u}_{\mathbf{P}}, t)), \quad \mathbf{X}_0, \mathbf{X} \in S, \tag{3}$$

with the correlation coefficient C and the displacement field $\mathbf{u}_{\mathbf{P}}(\mathbf{X}; \mathbf{P})$ defined over the subset S of the surface. The correlation coefficient C is often defined to have an extremum, typically a minimum, for the actual displacement field $\mathbf{u}(\mathbf{X}, t)$ in the neighborhood of $\mathbf{X}_0$. Thus, the problem of determination of the displacement field by image correlation in the neighborhood of a point $\mathbf{X}_0$ located within the subset S, may be stated as an optimization problem to find a set of local mapping parameters $\mathbf{P}^*$, such that

$$C(\mathbf{P}^*) = \min_{\mathbf{P} \in \mathbf{P}_D} C(\mathbf{P}), \tag{4}$$

where $\mathbf{P}_D$ is the domain of an assumed two-dimensional mapping function (cf., e.g. [7]).

A common correlation function is

$$C = \frac{\sum_S (f - g)^2}{\sum_S f^2}, \tag{5}$$

called the *least-square correlation* function. It is clear that it is non-negative and approaches 0 when f and g are close to each other.

E-mail:anand@mit.edu

[1] For an 8-bit system it is [0,255].

[2] It is also possible to obtain full three-dimensional displacement fields using more sophisticated imaging techniques, such as AFM.

Now, expanding the deformation at time t about $\mathbf{X}_0$, we obtain

$$\mathbf{y}(\mathbf{X},t) = \mathbf{y}(\mathbf{X}_0,t) + \mathbf{F}(\mathbf{X}_0,t)(\mathbf{X}-\mathbf{X}_0) + o(\mathbf{X}-\mathbf{X}_0), \quad (6)$$

where $\mathbf{F} = \partial\mathbf{y}/\partial\mathbf{X}$ is the deformation gradient. Thus, in the neighborhood of a point $\mathbf{X}_0$, and to within an error of $o(\mathbf{X}-\mathbf{X}_0)$, a deformation behaves like a homogeneous deformation with deformation gradient $\mathbf{F}(\mathbf{X}_0,t)$:

$$\mathbf{y}(\mathbf{X},t) = \mathbf{y}(\mathbf{X}_0,t) + \mathbf{F}(\mathbf{X}_0,t)(\mathbf{X}-\mathbf{X}_0). \quad (7)$$

This may be equivalently written in terms of the displacement and displacement gradient as

$$\mathbf{u}(\mathbf{X},t) = \mathbf{u}(\mathbf{X}_0,t) + \frac{\partial\mathbf{u}}{\partial\mathbf{X}}(\mathbf{X}_0,t)(\mathbf{X}-\mathbf{X}_0). \quad (8)$$

Hence, once the displacement vector field $\mathbf{u}(\mathbf{X},t)$ in the neighborhood of $\mathbf{X}_0$ at time t has been calculated, all other quantities, such as the deformation gradient $\mathbf{F}(\mathbf{X}_0,t)$, the stretch $\mathbf{U}(\mathbf{X}_0,t)$, the rotation $\mathbf{R}(\mathbf{X}_0,t)$, the finite Green strain $\mathbf{E}(\mathbf{X}_0,t)$ tensors at $(\mathbf{X}_0,t)$ are easily calculated using standard definitions:

$$\mathbf{F}(\mathbf{X}_0,t) = \mathbf{1} + \frac{\partial\mathbf{u}}{\partial\mathbf{X}}(\mathbf{X}_0,t), \quad (9)$$

$$\mathbf{U}(\mathbf{X}_0,t) = \sqrt{\mathbf{F}(\mathbf{X}_0,t)^\top\mathbf{F}(\mathbf{X}_0,t)}, \quad (10)$$

$$\mathbf{R}(\mathbf{X}_0,t) = \mathbf{F}(\mathbf{X}_0,t)\mathbf{U}(\mathbf{X}_0,t)^{-1}, \quad (11)$$

$$\mathbf{E}(\mathbf{X}_0,t) = \tfrac{1}{2}\left[\mathbf{F}(\mathbf{X}_0,t)^\top\mathbf{F}(\mathbf{X}_0,t) - \mathbf{1}\right]. \quad (12)$$

Writing (X_0, Y_0) for the in-plane cartesian components of $\mathbf{X}_0$, (X,Y) for the components of $\mathbf{X}$, (u_0, v_0) for the components of $\mathbf{u}(\mathbf{X}_0,t)$, (u,v) for the components of $\mathbf{u}(\mathbf{X},t)$, and ($\frac{\partial u}{\partial X}$, $\frac{\partial u}{\partial Y}$, $\frac{\partial v}{\partial X}$, $\frac{\partial v}{\partial Y}$) for the components of $\frac{\partial\mathbf{u}}{\partial\mathbf{X}}(\mathbf{X}_0,t)$, we may write (8) in expanded form as

$$u(X,Y) = u_0 + \frac{\partial u}{\partial X}(X-X_0) + \frac{\partial u}{\partial Y}(Y-Y_0), \quad (13)$$

$$v(X,Y) = v_0 + \frac{\partial v}{\partial X}(X-X_0) + \frac{\partial v}{\partial Y}(Y-Y_0). \quad (14)$$

In executing a DIC algorithm, the image of the reference configuration at time t_0 is divided into a mesh of subregions S. Most current DIC algorithms (cf., e.g., [2,7]) assume that the deformation field in each subregion S with respect to its own reference point $\mathbf{X}_0$ is homogeneous, and take the parameter set $\mathbf{P}$ as

$$\mathbf{P} = \left\{u_0, v_o, \frac{\partial u}{\partial X}, \frac{\partial v}{\partial X}, \frac{\partial u}{\partial Y}\frac{\partial v}{\partial Y}\right\}. \quad (15)$$

The minimization problem (4) for each subregion is then typically solved using a coarse search to find a rough estimate for the solution vector $\mathbf{P}^*$, and then the Newton-Raphson method is used to refine the solution. The n^{th} partial correction for the Newton-Raphson method is given by

$$\mathbf{\Delta P_n} = -\mathbf{H^{-1}}(\mathbf{P_n}) * \nabla\mathbf{C}(\mathbf{P_n}) \quad (16)$$

where in matrix notation,

$$\mathbf{\Delta P} = (dP_1, dP_2 \ldots dP_6)^T, \quad (17)$$

$$\mathbf{H} = \left[\frac{\partial^2 C}{\partial P_i \partial P_j}\right] \quad i,j = 1,2\ldots 6, \quad (18)$$

$$\nabla C(\mathbf{P}) = \left(\frac{\partial C}{\partial P_1}, \frac{\partial C}{\partial P_2} \cdots \frac{\partial C}{\partial P_6}\right)^T. \quad (19)$$

Calculation of the Hessian matrix $\mathbf{H}$ is the most time-consuming step in the Newton-Raphson method. In the commonly used method described above, the vector $\mathbf{P}$ has 6 components and the Hessian matrix $\mathbf{H}$ has 36 components. In the next section we describe our new algorithm, which substantially reduces the calculation time for estimating the local displacement fields within the various subregions S in a mesh, using standard ideas of interpolations from finite element techniques.

II. New Algorithm

Our new algorithm also follows a coarse-fine search approach. The major difference is how it describes the local displacement field while carrying out the Newton-Raphson calculations. In the algorithm discussed in the previous section, the displacement field $\mathbf{u}(\mathbf{X},t)$ in each subregion is determined locally; information from neighboring subregions is not utilized. However, when the deformation is continuous, the displacement field of each subregion has to satisfy compatibility requirements, and therefore the displacement fields of adjacent subregions are not totally independent.

Recall that in order to calculate the correlation function

$$C = \frac{\sum_S \left(f(\mathbf{X},t_0) - g(\mathbf{X}+\mathbf{u}(\mathbf{X}),t)\right)^2}{\sum_S f(\mathbf{X},t_0)^2}, \quad (20)$$

it is necessary to know the displacement field $\mathbf{u}(\mathbf{X},t)$ at all points $\mathbf{X}$ in S. In the new algorithm, this displacement field is determined by interpolating the displacement of the centroid $\mathbf{X}_0$ of S and the centeroid displacements of its eight surrounding subregions, see Fig. 1, as follows. Labelling the nine subregions as a 3×3 matrix, the region S of interest is the subregion '22'. Let the coordinates of a point $\mathbf{X}$ in subregion '22' at time t be denoted by $(X = X_0^{(22)} + \Delta X, Y = Y_0^{(22)} + \Delta Y)$, where $(X_0^{(22)}, Y_0^{(22)})$ are the coordinates of its centeroid $\mathbf{X}_0$. Then, the components of the displacement field in the subregion of interest are interpolated as

$$u^{(22)}(X,Y) = \sum_{i,j=1}^{3} h_{ij} u_0^{(ij)}, \quad (21)$$

$$v^{(22)}(X,Y) = \sum_{i,j=1}^{3} h_{ij} v_0^{(ij)}, \quad (22)$$

where

$$\left\{u_0^{(ij)}, v_0^{(ij)}\right\}, \quad i,j = 1,2,3, \quad (23)$$

denote the components of the displacement of the centroids of the (ij) th subregion in a 3×3 patch of nine subregions,

as depicted in Fig. 1, and h_{ij} are interpolation functions given by

$$h_{11} = \frac{1}{4}r(1-r)s(1-s), \quad (24)$$
$$h_{12} = -\frac{1}{2}(1-r^2)s(1-s), \quad (25)$$
$$h_{13} = -\frac{1}{4}r(1+r)s(1-s), \quad (26)$$
$$h_{21} = -\frac{1}{2}r(1-r)(1-s^2), \quad (27)$$
$$h_{22} = (1-r^2)(1-s^2), \quad (28)$$
$$h_{23} = \frac{1}{2}r(1+r)(1-s^2), \quad (29)$$
$$h_{31} = -\frac{1}{4}r(1-r)s(1+s), \quad (30)$$
$$h_{32} = \frac{1}{2}(1-r^2)s(1+s), \quad (31)$$
$$h_{33} = \frac{1}{4}r(1+r)s(1+s), \quad (32)$$

where the local coordinates (r, s) are calculated as

$$r = \Delta X/L, \quad (33)$$
$$s = \Delta Y/L, \quad (34)$$

with L denoting the distance between the centers of two adjacent subregions.

We take the parameter set $\mathbf{P}$ *for a subregion* S *as the components of the displacement vector* (u_0, v_0) *of its centroid* $\mathbf{X}_0$, and continuing with the nomenclature above for the subregion '22',

$$\mathbf{P} = \left\{u_0 \equiv u_0^{(22)}, v_0 \equiv v_0^{(22)}\right\}. \quad (35)$$

With this parameter set, the n^{th} partial corrections in the Newton-Raphson method are again given by

$$\Delta\mathbf{P_n} = -\mathbf{H^{-1}}(\mathbf{P_n}) * \nabla\mathbf{C}(\mathbf{P_n}), \quad (36)$$

but now

$$\Delta\mathbf{P} = (du_0, dv_0)^T, \quad (37)$$
$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 C}{\partial u_0^2} & \frac{\partial^2 C}{\partial u_0 \partial v_0} \\ \frac{\partial^2 C}{\partial u_0 \partial v_0} & \frac{\partial^2 C}{\partial v_0^2} \end{bmatrix}, \quad (38)$$
$$\nabla C(\mathbf{P}) = \left(\frac{\partial C}{\partial u_0}, \frac{\partial C}{\partial v_0}\right)^T. \quad (39)$$

Thus, we see that in the new algorithm the parameter set $\mathbf{P}$ and the Hessian matrix $\mathbf{H}$ have only 2 and 4 components, respectively, compared with 6 and 36 for the algorithm defined in the previous section. Further, we expect that the displacement field described by (21) and (22) will be a better representation of the displacement field described by (13) and (14) when the deformation is non-homogeneous, because the displacement fields of neighboring subregions are related to each other, and this information is being utilized in the new algorithm.

As an example of one of the derivatives for the Newton-Raphson procedure we note that one has to calculate derivatives of the type

$$\frac{\partial C}{\partial u_0} = -\frac{2}{\sum_S f(\mathbf{X}, t_0)^2} \sum_S \Big[\Big(f(\mathbf{X}, t_0) - g(\mathbf{X} + \mathbf{u}(\mathbf{X}), t)\Big) \times \frac{\partial g(\mathbf{X} + \mathbf{u}(\mathbf{X}), t)}{\partial u_0}\Big], \quad (40)$$

where we recall that at any given time t, g is a two-dimensional function of (x, y). Since $g(x, y)$ is an image file, or a two-dimensional matrix with a value only when x and y are integers. Thus, to facilitate the evaluation of $g(\mathbf{x}, t)$ and its derivatives at subpixel locations during the solution process, it is necessary to interpolate the intensity values of g between pixels.

Typically, a bicubic spline interpolation is used for interpolating the intensity values of g between pixels. In contrast, here we use a piecewise polynomial interpolation for the intensity. With respect to Fig. 2, the value of g in the gray-colored region of interest may be interpolated as,

$$g(p, q) = \sum_{i,j=-1}^{2} h_{(p,i)} h_{(q,j)} g(i, j), \qquad p, q \in [0, 1], \quad (41)$$

where $g(i, j)$ are the intensities at the node points labeled in Fig. 2, and $h_{(p,i)}$ and $h_{(q,j)}$ are the following interpolation functions

$$h_{(p,-1)} = -\frac{1}{6}p(p-1)(p-2), \quad (42)$$
$$h_{(p,0)} = \frac{1}{2}(p+1)(p-1)(p-2), \quad (43)$$
$$h_{(p,1)} = -\frac{1}{2}(p+1)p(p-2), \quad (44)$$
$$h_{(p,2)} = \frac{1}{6}(p+1)p(p-1), \quad (45)$$
$$h_{(q,-1)} = -\frac{1}{6}q(q-1)(q-2), \quad (46)$$
$$h_{(q,0)} = \frac{1}{2}(q+1)(q-1)(q-2), \quad (47)$$
$$h_{(q,1)} = -\frac{1}{2}(q+1)q(q-2), \quad (48)$$
$$h_{(q,2)} = \frac{1}{6}(q+1)q(q-1). \quad (49)$$

Given this form of the intensity interpolation, the first order spatial directives of the intensity are readily derived, and a typical derivative is listed below:

$$\frac{\partial g}{\partial u_0} = \frac{\partial g}{\partial p} = \sum_{i,j=-1}^{2} \frac{\partial h_{(p,i)}}{\partial p} h_{(q,j)} g(i, j). \quad (50)$$

III. Evaluation of the new algorithm

In this section we evaluate the accuracy of the new algorithm by checking its results against *numerically-generated*

images which are subjected to two different deformation histories.

In order to use the DIC method, the surface of a specimen should either exhibit suitable natural random patterns, or it should be prepared by painting it with dots of random shapes, appropriate sizes and frequencies. It is possible to carry out such a surface preparation procedure digitally to create an artificial image to represent such a prepared surface. Further, such an image can also be manipulated digitally to impose any desired deformation field. The advantage of creating a sequence of deformed images digitally, is that the imposed deformation field can be of any form and precisely controlled, and any noise associated with imaging hardware and other sources of errors in physical experiments are eliminated.

The following two-dimensional polynomial function was used to generate a single two-dimensional 'dot',

$$z = 1 - \frac{1}{c}\left[a \times (x - x_0)^2 + b \times (y - y_0)^2\right]^m. \qquad (51)$$

Here, the parameters a and b are random numbers between $[0.9, 1.1]$ which give each dot a different ratio between its dimensions in the x and y directions; the parameter m controls the shape of the dot; and c controls the size of the dot for a specific m. The reason for choosing random numbers for a and b is to make every dot distinct, so that later in the DIC calculation, the displacement solutions for the subregions will be reasonably different. This function has a value z in the range $(-\infty, 1]$. To convert a value of z for an image file, we need to multiply it by a suitable number, such as 255 for an 8-bit gray-scale image. Then, if the result is positive, its integer part is used in the image file, and if the result is negative, its value in the image file is set to zero. Fig. 3 illustrates the profile of such a dot and its image form. The values $m = 2$ and $c = 500$ were used to create this dot; this combination gives us a radius of roughly 5 pixels. A desired mesh of dots can be generated by successively repeating the procedure for a single dot.

A 'deformation' of the 2D polynomial function (51) can be achieved by transforming the original function. For example, the transformation

$$z = 1 - \frac{1}{c}\left[a \times (\lambda \cdot (x - \Delta x - x_0))^2 + b \times (y - y_0)^2\right]^m, \qquad (52)$$

is equivalent to the surface having a rigid body motion Δx and a stretch λ in the x direction. Shear, rotation or any other kind of deformations can be imposed in a similar fashion. Two different types of deformation fields have been used to test the accuracy of our DIC algorithm.

The first is a homogeneous simple shear. Fig. 4 a shows the undeformed mesh of points representing a surface, and Fig. 4 b shows the deformed geometry after a simple shear

$$x = X + \gamma Y, \quad y = Y, \qquad X, Y \in [0, 1], \qquad (53)$$

with $\gamma = 0.2$. The deformation gradient tensor $\mathbf{F}$ for this homogeneous deformation was calculated, and the polar decomposition of $\mathbf{F}$ was performed to get the rotation $\mathbf{R}$ and the stretch $\mathbf{U}$, together with principal values (λ_1, λ_2). Ten sequential frames were used to analyze the deformation. Fig. 4 c shows a comparison of the principal stretches calculated using the DIC method against those calculated from the numerically imposed deformation (53), and Fig. 4 d shows the corresponding calculations for the rotation, which for this two-dimensional deformation may be represented by an angle θ. In these figures, the full lines represent the theoretical value of the imposed deformation field, and the points are obtained from the new DIC algorithm. The results from the DIC calculation are in excellent agreement with those from the numerically-imposed deformation.

The simple non-homogeneous deformation field tested here is

$$x = (1 + \epsilon)\left(X - \frac{1}{2}\right) + \frac{1}{2}, \quad y = Y, \quad X, Y \in [0, 1], \qquad (54)$$

$$\epsilon = \epsilon(Y) = \epsilon_0 \left(4\Big(Y - \frac{1}{2}\Big)^2 - 1\right), \quad \epsilon_0 = 0.1, \qquad (55)$$

with the following components for the two-dimensional *infinitesimal strain measure*,

$$[\boldsymbol{\epsilon}] = \begin{bmatrix} \epsilon & \frac{1}{2}\,\dfrac{\partial \epsilon}{\partial Y}\,X \\ \frac{1}{2}\,\dfrac{\partial \epsilon}{\partial Y}\,X & 0 \end{bmatrix}. \qquad (56)$$

Fig. 5a shows the undeformed mesh of points representing a surface, and Fig. 5b shows the deformed geometry. Fig. 5c shows the variation of the numerically-imposed strain $\epsilon_{xx} = \epsilon(Y)$ and compares it to the data calculated using the new DIC algorithm; as before, the solid line represents the theoretical value and the points are obtained from the DIC algorithm. We see that the DIC result matches the imposed strain variation very well.

Finally, Figs. 6a,b show the initial and deformed geometries for a mesh subjected to a similar non-homogeneous deformation, but this time for only one-quarter of the specimen. Figs. 6c,d show the corresponding distributions of ϵ_{XX} and ϵ_{XY} calculated using the DIC method. These distributions also match the theoretically-imposed ones, but the latter are not plotted in Figs. 6c,d for clarity.

IV. Experimental results

First we demonstrate the capability of our DIC method in a large strain application of determining the axial stretch during simple extension of a rubbery polymer. Because of the typical inhomogeneity of the deformation that develops in such experiments near the grips, the cross-head extension cannot be used to calculate the actual strain in the gauge section. Further, since the material is relatively soft and subjected to large strains, mechanical extensometers are also not suitable for strain measurements in such experiments. However, the non-contact DIC method may be conveniently used in this application.

Fig. 7a shows the initial surface of a tensile specimen of this material; the *gauge length* for calculating the axial stretch is also indicated in this figure. This time the surface

of the specimen is physically painted with a mesh of dots. The tensile test is carried out under displacement control. An image was taken every second, while the axial load F was also recorded at the same frequency. Fig. 7b shows the specimen after a maximum axial stretch of $\lambda = 2.25$, which corresponds to an engineering strain of 125%. After processing the images with the new DIC method we obtained the stretch versus time curve $\lambda(t)$; combining it with the load versus time curve $F(t)$, we obtained the load-stretch curve $F(\lambda)$ shown in Fig. 7c. From the load-stretch data, and knowledge about the specimen geometry, and the standard assumption of incompressibility for rubber-like materials, other quantities such as true stress and true strain are easily calculated.

As a second example, we present results from an effort to determine the *incremental strain distribution* beneath a sharp indenter impressed on another rubbery polymer. The indentation was conducted in a scanning electron microscope (SEM) and the high-magnification images before and after the deformation were obtained by using this instrument, Fig. 8a,b. By comparing these two images using the new DIC method, the strain field in the polymeric material under the indenter was calculated. *In this test, the surface of the specimen was not specially treated*; the natural features on the surface of the polymeric material provided enough information for us to use the DIC method. Figs. 8c,d, respectively, show the calculated contours of the strain components E_{XX} and E_{YY} of the finite Green strain measure $\mathbf{E}$; These figures clearly illustrate the concentration of the strain field underneath the indenter. The scale bar in Fig. 8a shows that the total region of interest here is only a few hundred square microns; it is difficult to measure the strain distribution in such a small region using any other current experimental method. However, the DIC method, combined with the SEM digital imaging capability, is readily useful for such an application at the microscale.

V. Concluding remarks

We have developed a new, reasonably fast, DIC algorithm for non-contact, two-dimensional, whole-field displacement and strain measurement. We have used numerically-generated digital images to show that the new algorithm accurately reproduces imposed homogeneous and non-homogeneous strain fields. The algorithm has also been used to measure the strain distributions from actual images obtained from deformed specimens, one at the macroscale of the stretch measurement of a tension test in an elastomeric material, and another at the microscale for the sharp indentation of another elastomeric material carried out in a SEM. In these two different challenging applications, the new DIC algorithm demonstrates some of its versatile attributes, such as a non-contact capability to measure large strains at the macroscale, and whole-field strain measurements at the microscale.

We have not carried out an exhaustive comparative study of the speed and accuracy of our algorithms versus existing algorithms for whole-field strain measurements in the literature. We leave such a study for future research.

Acknowledgments

The authors acknowledge the financial support provided by the Singapore-MIT Alliance, as well as ONR Contract N00014-01-1-0808 with MIT. The gracious help of Professor D. P.Hart of MIT in getting us started with digital image correlation methods is gratefully acknowledged.

References

[1] W. H. Peters, and W. F. Ranson, *Optical Engineering*, 1982, **21**, 427.
[2] M. A. Sutton, M. Cheng, W. H. Peters, Y. J. Chao, and S. R. McNeill, *Image and Vision Computing*, 1986, **4(3)**, 143.
[3] M. A. Sutton, S. R. McNeill, J. Jang, and M. Babai, *Optical Engineering*, 1988, **27(10)**, 870.
[4] H. A. Brucks, S. R. McNeill, M. A. Sutton, and W. H. Peters, *Experimental Mechanics*, 1989, **29**, 261.
[5] S. R. McNeill, M. A. Sutton, Z. Miao, and J. Ma, *Experimental Mechanics*, 1996, **36**, 13.
[6] G. Vendroux, and W. G. Knauss, *Experimental Mechanics*, 1998, **38(2)**, 86.
[7] W. Tong, *Experimental Mechanics*, 1997, **37(4)**, 452.
[8] E. Mazza, and J. Dual, *Journal of the Mechanics and Physics of Solids*, 1999, **47**, 1975.
[9] Y. Wang, G. Gioia, and A. M. Cuitino, *Journal of Engineering Materials and Technology*, 2000, **122**, 376.

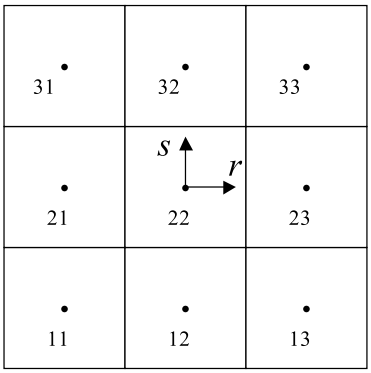


Fig. 1. Local coordinate system for the displacement interpolation.

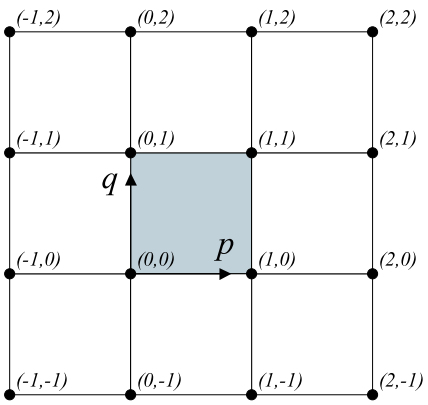
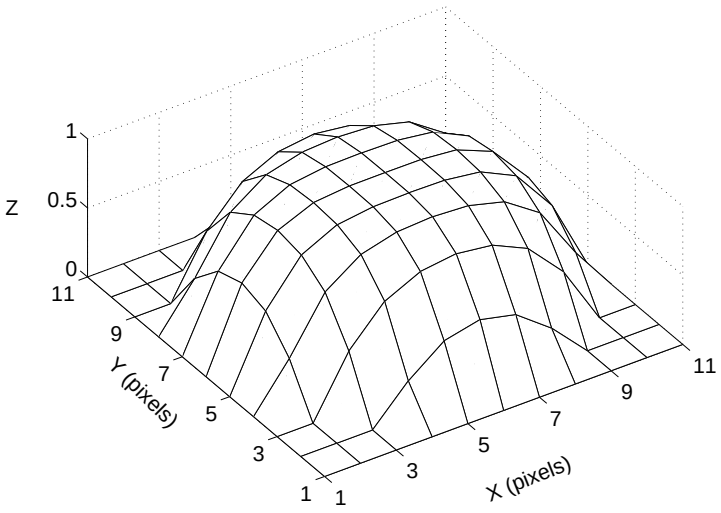
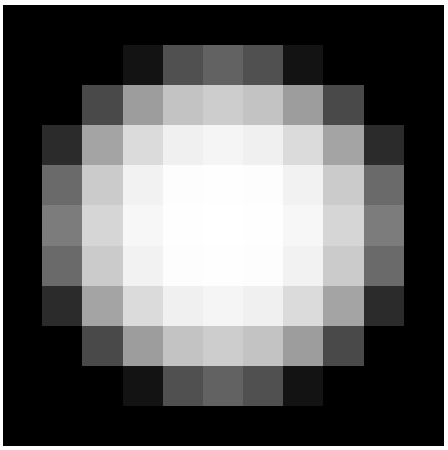


Fig. 2. Local coordinate system for the digital image intensity interpolation between pixels.

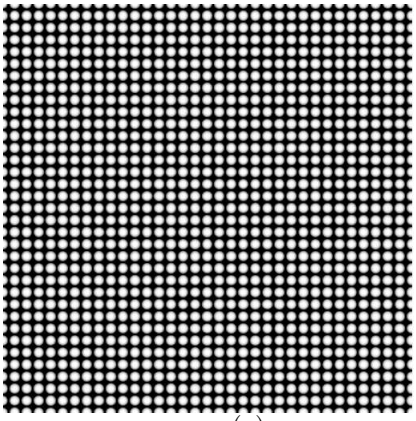


(a)

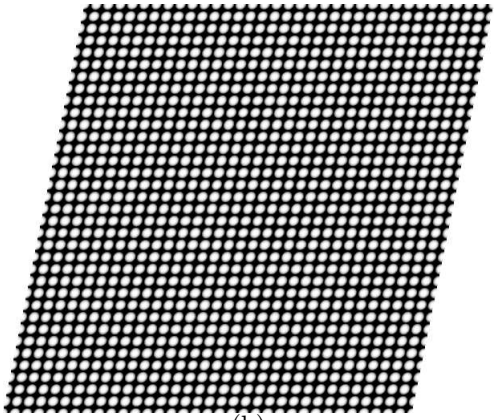


(b)

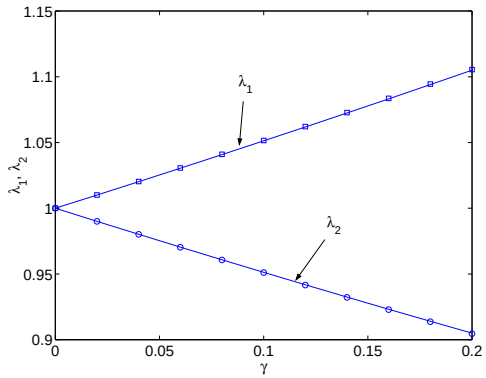
Fig. 3. Artificially created image: (a) profile of a single dot; (b) image of a single dot.



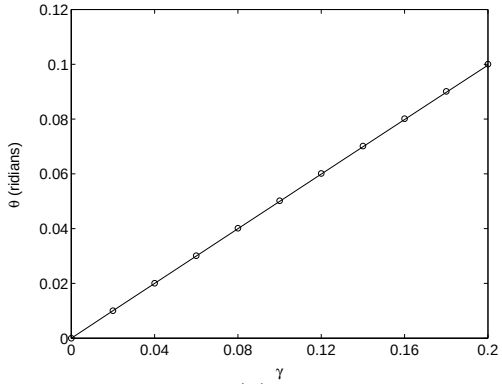
(a)



(b)

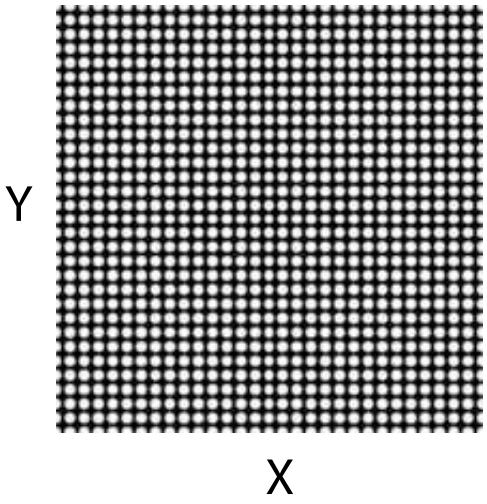


(c)

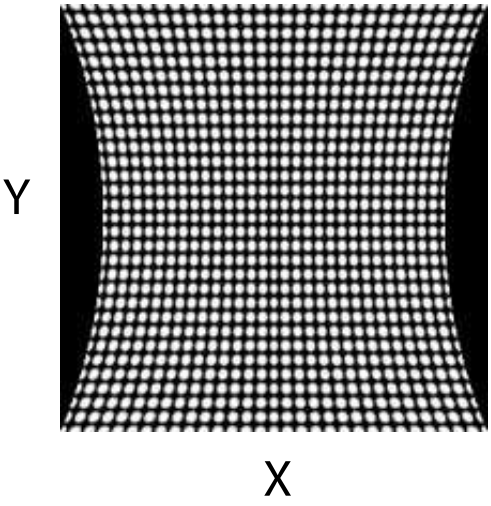


(d)

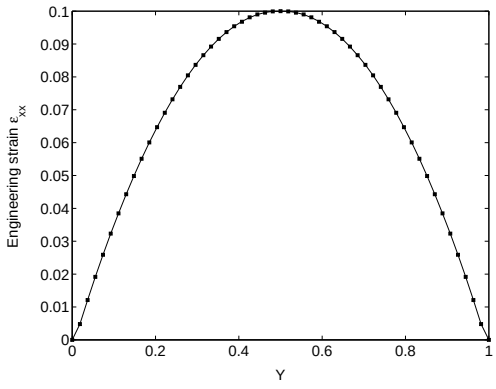
Fig. 4. Homogeneous simple shear deformation: (a) initial image; (b) deformed image; (c) variation of principal stretches λ_1, λ_2, with increasing amount of shear ; (d) variation of angle θ describing the rotation $\mathbf{R}$ in simple shear, with increasing amount of shear.



(a) Initial image

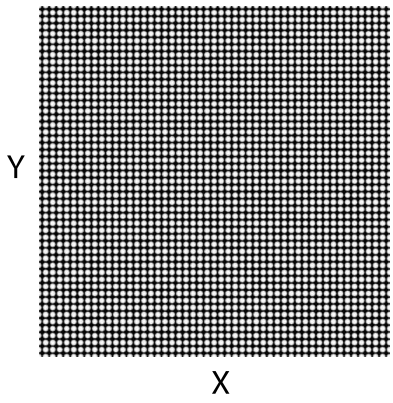


(b) Deformed image

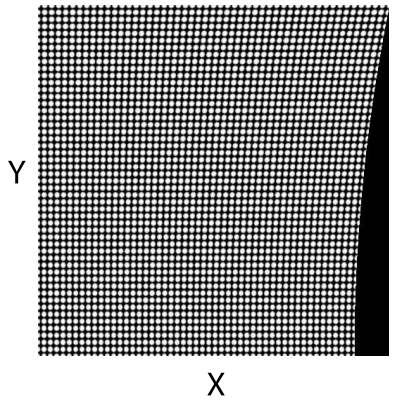


(c) ϵ_{XX}

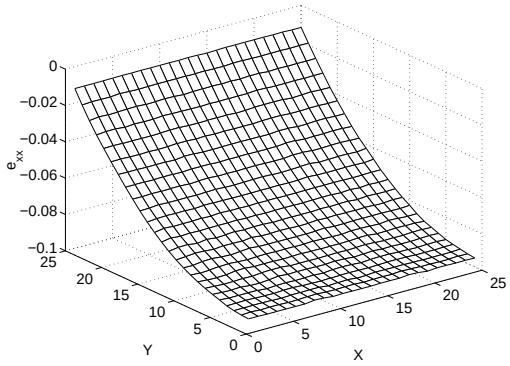
Fig. 5. Non-homogeneous deformation: (a) initial image; (b) deformed image; (c) distribution of strain ϵ_{XX} versus Y in the non-homogeneous deformation field.



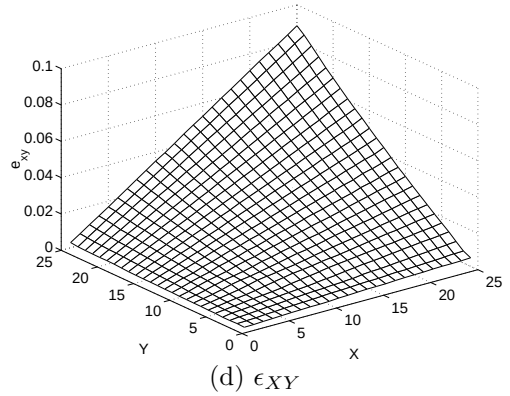
(a) Initial image



(b) Deformed image

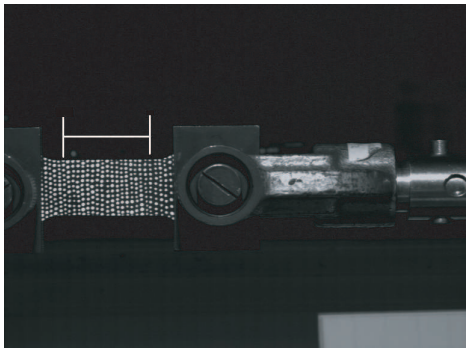


(c) ϵ_{XX}

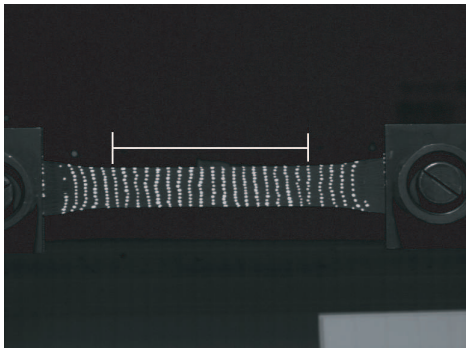


(d) ϵ_{XY}

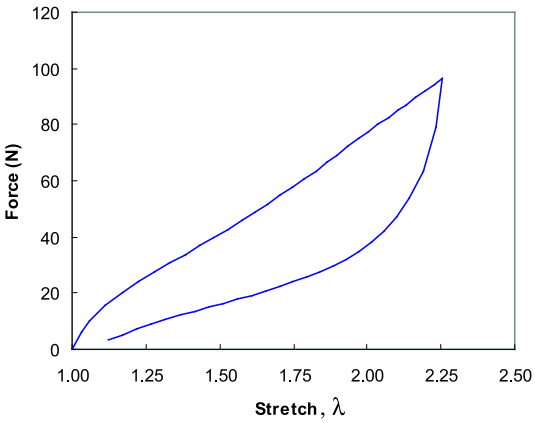
Fig. 6. Non-homogeneous deformation: (a) initial image; (b) deformed image; (c) plot of distribution of strain ϵ_{XX} versus X and Y; (c) plot of distribution of strain ϵ_{XY} versus X and Y.



(a) Initial image

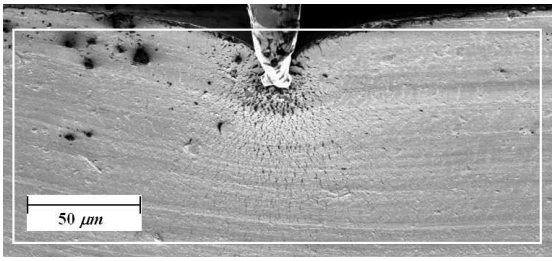


(b) Deformed image

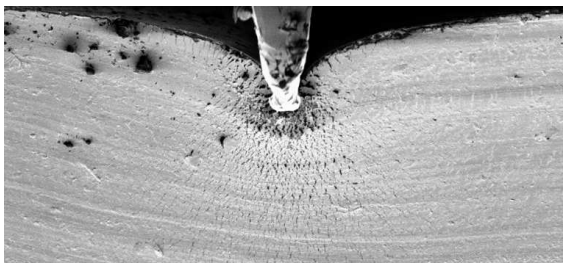


(c) stretch vs. force

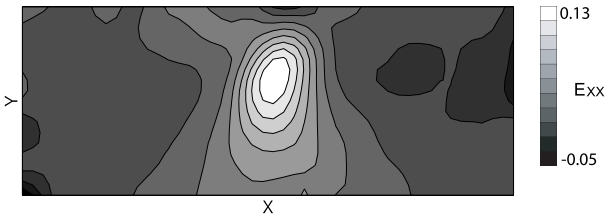
Fig. 7. Tensile test of a rubber specimen to a large strain: (a) initial image showing the gauge length used to measure the axial stretch; (b) final image; (c) calculated force versus stretch curve.



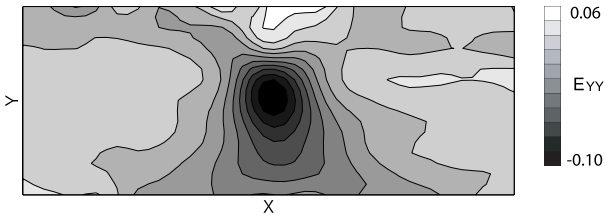
(a)



(b)



(c)



(d)

Fig. 8. Strain distribution in microscale sharp indentation of a rubbery polymer: (a) Initial image showing the white-framed area in which the incremental strain distribution is to be determined; (b) final deformed image; the images are obtained in a SEM. (c,d) Calculated contours of the E_{XX} and E_{YY} components, respectively, of the finite Green strain measure.